1. Write a program to input two matrices and do Matrix addition and subtraction

#include <stdio.h>

//Matrix addition and subtraction

int main()

{

int matrix1[3][3], matrix2[3][3];

int result[3][3]; //for printing addition and subtraction

int i, j;

printf("Enter matrix elements for first matrix: \n");

for (i = 0; i < 3; i++)

{

for ( j = 0; j < 3; j++)

{

printf("Enter element at matrix1[%d][%d]", i+1, j+1);

scanf("%d", &matrix1[i][j]);

}

}

printf("Enter matrix elements for Second matrix: \n");

for (i = 0; i < 3; i++)

{

for ( j = 0; j < 3; j++)

{

printf("Enter element at matrix2[%d][%d]", i+1, j+1);

scanf("%d", &matrix2[i][j]);

}

}

//Printing First Matrix

printf("First matrix:\n");

for (i = 0; i < 3; i++)

{

for ( j = 0; j < 3; j++)

{

printf("%5d", matrix1[i][j]);

}

printf("\n");

}

//Printing Second Matrix

printf("Second matrix:\n");

for (i = 0; i < 3; i++)

{

for ( j = 0; j < 3; j++)

{

printf("%5d", matrix2[i][j]);

}

printf("\n");

}

//Matrix Addition

printf("Matrix after Addition:\n");

for (i = 0; i < 3; i++)

{

for ( j = 0; j < 3; j++)

{

result[i][j] = matrix1[i][j] + matrix2[i][j];

printf("%5d", result[i][j]);

}

printf("\n");

}

//Matrix Subtraction

printf("Matrix after Subtraction:\n");

for (i = 0; i < 3; i++)

{

for ( j = 0; j < 3; j++)

{

result[i][j] = matrix1[i][j] - matrix2[i][j];

printf("%5d", result[i][j]);

}

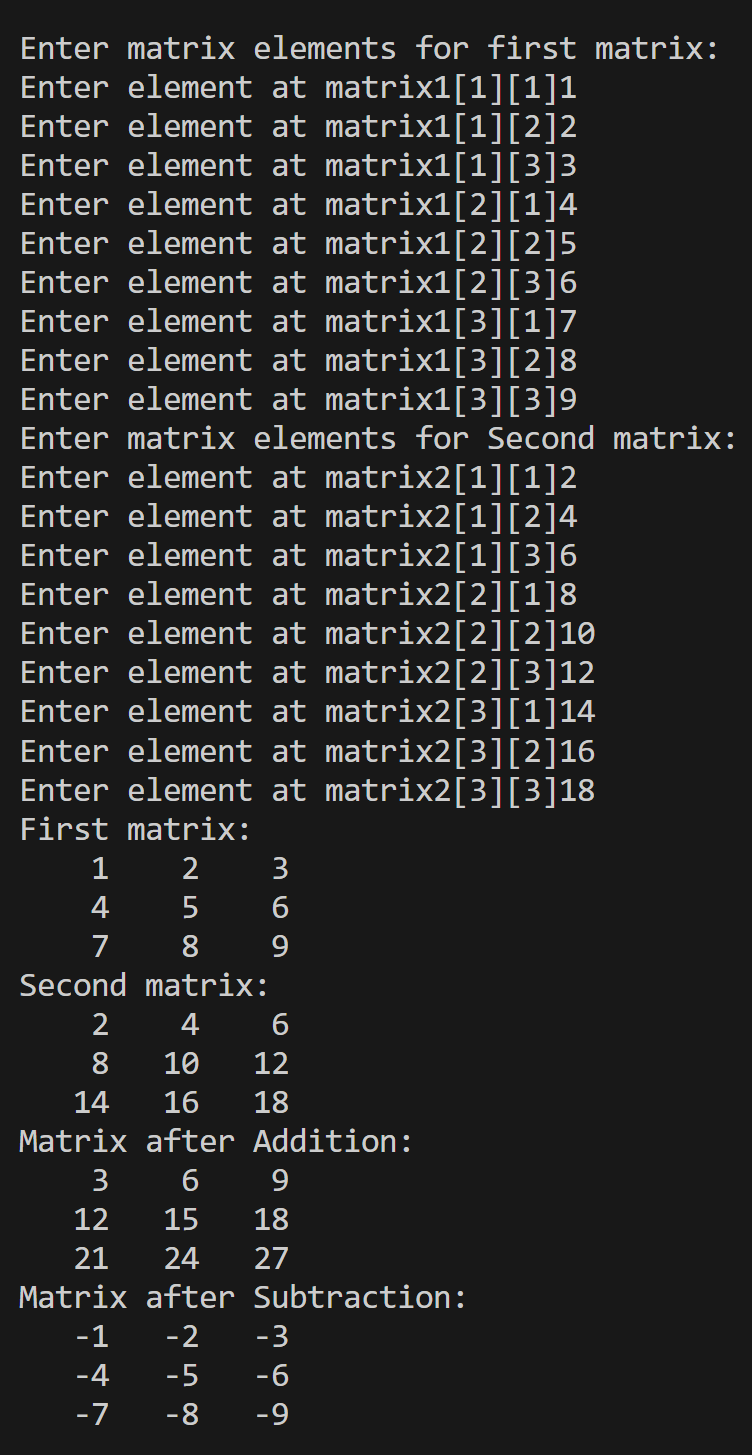
printf("\n");

}

return 0;

}

Output:



2. Write a program to calculate the Gravitational Force between two objects, taking G as 6.67 \* 10-11.

#include <stdio.h>

#define G 6.67e-11

double gravitational\_force(double m1, double m2, double d)

{

return G\*m1\*m2/(d\*d);

}

int main()

{

double m1, m2, d, force;

printf("Enter mass of first object (in kg) ");

scanf("%lf", &m1);

printf("Enter mass of second object (in kg) ");

scanf("%lf", &m2);

printf("Enter distance between objects (in meters) ");

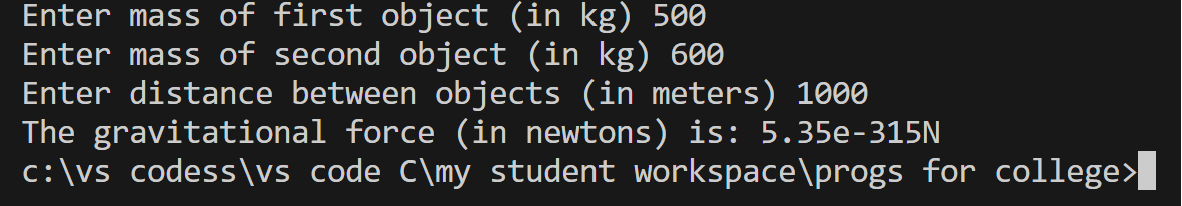
scanf("%lf", &d);

force = gravitational\_force(m1,m2,d);

printf("The gravitational force (in newtons) is: %.2eN");

return 0;

}

Output:

3. Write a C program to check if a given number remains the same even after reversing its digits(without using string functions). These types of numbers are also called palindrome numbers. For example, 121, 1331, 2468642 are these types of numbers, but 123, 145, 9876 are not.

#include <stdio.h>

int main()

{

int n, reversed = 0, remainder;

printf("This program is to check whether a number is palindrome or not \nEnter the number you want to check ");

scanf("%d", &n);

int original = n; // because we'll do manupulations on n so we need original number saved

while (n != 0)

{

remainder = n % 10; // to get the last digit

reversed = reversed \* 10 + remainder; // to keep appending digits

n = n / 10;

}

if (reversed == original)

{

printf("Yes, %d is a palindrome number", original);

}

else

{

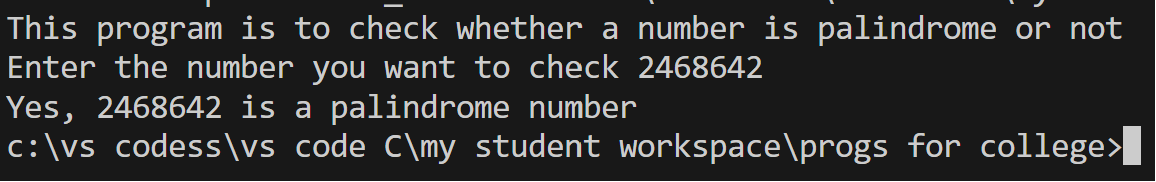
printf("No, %d is not a palindrome number", original);

}

return 0;

}

Output:



4. Write a program to make a currency converter, that can convert minimum 3 currencies into one another(say USD, EURO & INR). Keeping prices fixed and ignoring real time change.

#include <stdio.h>

//1USD = 80INR, 1EURO = 85INR, 1EURO= 1.06USD

int main()

{

int choice;

float rupees, dollars, euros;

printf("This is a currency converter\n");

printf("1.Rupees to Dollars \n");

printf("2.Dollars to Rupees \n");

printf("3.Rupees to Euro \n");

printf("4.Euro to Rupees \n");

printf("5.Dollars to Euros \n");

printf("6.Euros to Dollars \nEnter your choice ");

scanf("%d", &choice);

switch (choice)

{

case 1: //Rupees to Dollars

printf("Enter amount in Rupees ");

scanf("%f", &rupees);

dollars = rupees/80;

printf("Amount in Dollars %.2f $", dollars);

printf("\nNote: This program works on fixed prices. Real time prices may differ");

break;

case 2: //Dollars to Rupees

printf("Enter amount in Dollars ");

scanf("%f", &dollars);

rupees = dollars\*80;

printf("Amount in Rupees %.2f ", rupees);

printf("\nNote: This program works on fixed prices. Real time prices may differ");

break;

case 3: //Rupees to Euro

printf("Enter amount in Rupees ");

scanf("%f", &rupees);

euros = rupees/85;

printf("Amount in euros %.2f ", euros);

printf("\nNote: This program works on fixed prices. Real time prices may differ");

break;

case 4: //Euro to Rupees

printf("Enter amount in Euros ");

scanf("%f", &euros);

rupees = euros\*85;

printf("Amount in Rupees %.2f ", rupees);

printf("\nNote: This program works on fixed prices. Real time prices may differ");

break;

case 5: //Dollars to Euros

printf("Enter amount in Dollars ");

scanf("%f", &dollars);

euros = dollars/1.06;

printf("Amount in Euros %.2f ", euros);

printf("\nNote: This program works on fixed prices. Real time prices may differ");

break;

case 6: //Euros to Dollars

printf("Enter amount in euros ");

scanf("%f", &euros);

dollars = euros\*1.06;

printf("Amount in Dollars %.2f $", dollars);

printf("\nNote: This program works on fixed prices. Real time prices may differ");

break;

default:

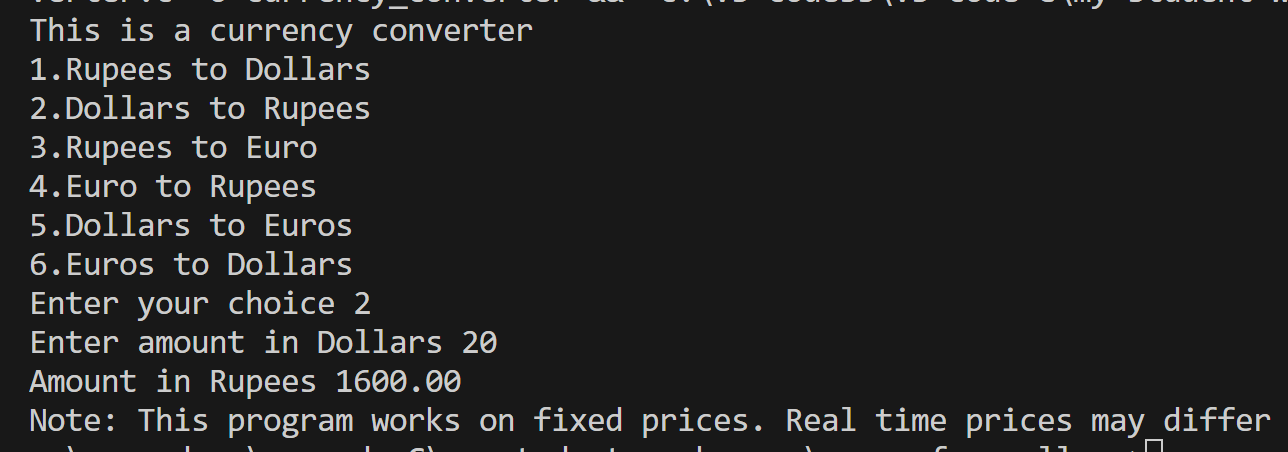
printf("Wrong input");

break;

}

return 0;

}

Output:

5. Write a program that takes user input n and prints numbers which are the cube roots of the product of their proper divisors upto n. For example 24 = 2,3,4,6,8,12, their product is 13,824 whose cube root is again 24.

#include <stdio.h>

int productOfDivisors(int num)

{

int product = 1;

for (int i = 2; i <= num / 2; i++) {

if (num % i == 0) {

product \*= i;

}

}

return product;

}

int main() {

int n;

printf("Enter a value for n: ");

scanf("%d", &n);

printf("Numbers up to %d whose cube roots are equal to the product of their proper divisors:\n", n);

for (int i = 1; i <= n; i++)

{

int cube = i \* i \* i;

int divisorProduct = productOfDivisors(i);

if (cube == divisorProduct)

{

printf("%5d ", i);

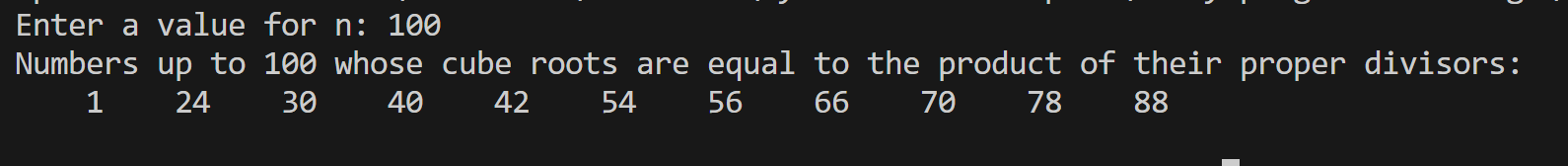
}

}

printf("\n");

return 0;

}

Output:

6. Write a program that asks a user input but it keeps asking input until he enters 0. Print the product of all user input numbers and number of times user did input a number.

#include <stdio.h>

int main()

{

int n, product = 1, count = 0;

printf("Give me a number and I will end this program ");

scanf("%d", &n);

product = product\*n;

while (n!=0)

{

printf("Crap, try again ");

scanf("%d", &n);

if (n!=0)

{

product = product\*n;

count +=1;

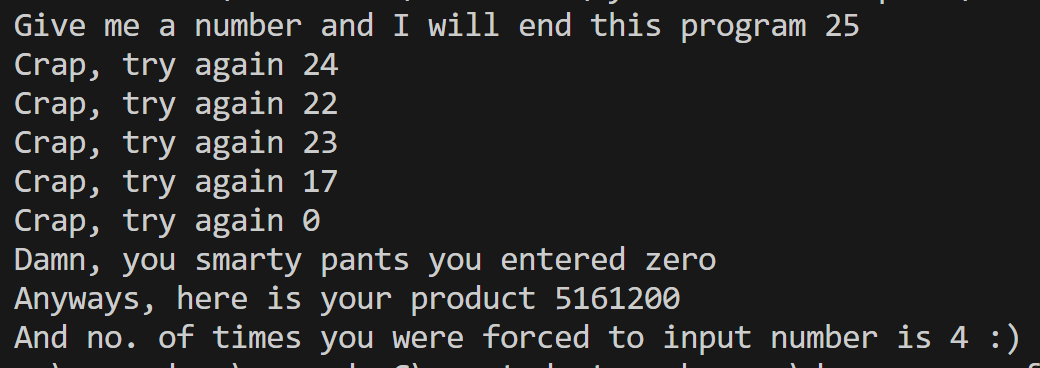
}

}

printf("Damn, you smarty pants you entered zero\nAnyways, here is your product %d\nAnd no. of times you were forced to input number is %d :)", product, count);

return 0;

}

Output:

7. Write a program to merge two arrays into a third array.

#include <stdio.h>

int main()

{

int size\_one, size\_two; //size of array

int count = 0, tout = 0;

printf("Enter size of array 1 ");

scanf("%d", &size\_one);

printf("Enter size of array 2 ");

scanf("%d", &size\_two);

int i, j, k, m, n;

int tot\_size; //size of final array

int arr1[size\_one], arr2[size\_two];

//First array input loop

printf("Input elements in first array\n");

for (i = 0; i < size\_one; i++)

{

printf("Enter elements for arr1[%d]", i);

scanf("%d", &arr1[i]);

}

//Second array input loop

printf("Input elements in Second array\n");

for (k = 0; k < size\_two; k++)

{

printf("Enter elements for arr2[%d]", k);

scanf("%d", &arr2[k]);

}

//Merging in final array

tot\_size = size\_one + size\_two;

int finarr[tot\_size]; //final array declare

for (j = 0; j < size\_one; j++)

{

finarr[j] = arr1[j]; //merging first into final

}

for (m = size\_one; m < tot\_size; m++)

{

finarr[m] = arr2[m-size\_one]; //merging second into final

}

int pout = 0;

printf("Merged array:\n");

for (n = 0; n < tot\_size; n++)

{

printf("%5d", finarr[n]);

if (n==size\_one-1)

{

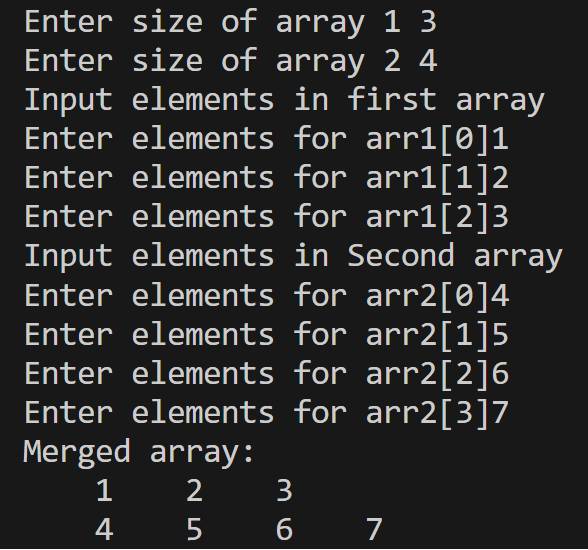
printf("\n"); //to print in new line for elements for second array that are now merged

}

}

return 0;

}

Output:

8. Create a structure of Student. Input details like roll number, name, and and marks of 3 subjects(phy, chem, math). Show the name, marks of 3 subjects and also the total in a bit organized way.

#include <stdio.h>

#include <string.h>

struct student

{

char name[30];

int phy\_marks;

int mat\_marks;

int che\_marks;

int roll;

};

int total;

int main()

{

int i, j, n; // n for number of students

// i & j for counter

printf("Hii, This program is to input student details, marks and get their total\nEnter the no. of students you want to input details of");

scanf("%d", &n);

struct student st[n]; // making array of student datatype that contains n students

for (i = 0; i < n; i++)

{

printf("Enter name of student[%d] ", i + 1);

scanf("%s", st[i].name);

printf("Enter roll number of student[%d] ", i + 1);

scanf("%d", &st[i].roll);

printf("Enter marks of physics of student[%d] ", i + 1);

scanf("%d", &st[i].phy\_marks);

printf("Enter marks of mathematics of student[%d] ", i + 1);

scanf("%d", &st[i].mat\_marks);

printf("Enter marks of chemistry of student[%d] ", i + 1);

scanf("%d", &st[i].che\_marks);

}

printf("Details of each student below:\n");

printf("Name \t\tMarks\n");

for (i = 0; i < n; i++)

{

printf("%s \t\n", st[i].name);

printf(" \t\t%d(Physics)\n", st[i].phy\_marks);

printf(" \t\t%d(Mathematics)\n", st[i].mat\_marks);

printf(" \t\t%d(Science)\n", st[i].che\_marks);

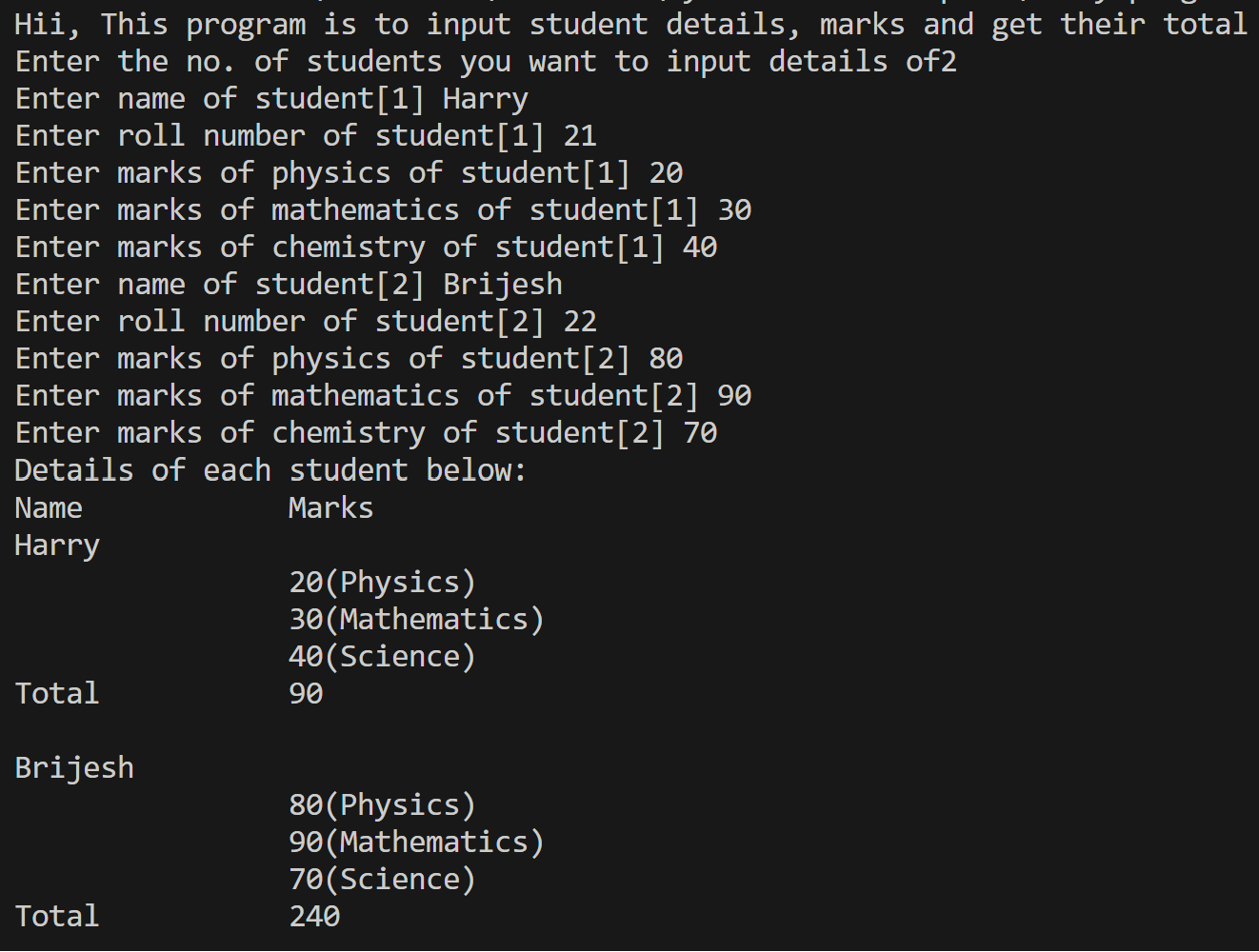
total = st[i].phy\_marks + st[i].mat\_marks + st[i].che\_marks;

printf("Total \t\t%d\n\n", total);

}

return 0;

}

Output:

9. Write a program to know the day of a particular date.

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <math.h>

int main()

{

int day, month, year;

printf("Enter date in DD MM YYYY format ");

scanf("%d %d %d", &day, &month, &year);

int daysinmonth[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

if(month==2)

{

if((year % 4 ==0 && year % 100 != 0) || (year % 400 == 0))

{

daysinmonth[2] = 29;

}

}

char\* weekdays[] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}; //0 = Sunday, 6 = Saturday

if (day<1 || day > 31 || month<1 || month > 12 || year<1950 || year > 9999)

{

printf("Invalid input ");

}

else{

int totaldays = 0;

for (int y = 1950; y < year; y++)

{

totaldays += 365;

if((y % 4 == 0 && y % 100 != 0) || (y % 400 == 0))

{

totaldays += 1;

}

}

for (int m = 1; m < month; m++)

{

totaldays += daysinmonth[m];

}

totaldays += day - 1; // subtracting 1 because days start with 1 so it doesn't add 2 for 2nd of Jan, 1950

int weekdaysindex = totaldays % 7;

// printf("weekdaysindex = %d", weekdaysindex);

printf("The day is %s", weekdays[weekdaysindex]);

}

return 0;

}

Output: